

# 情報処理概論(第6回)

情報基盤センター 井上 仁

## 今日の授業内容(第6回)

- DO構文(続き)
- 配列
- DO形並び

## DO構文の復習(1)

- n回繰り返す
- ある条件を満たす間繰り返す

<pre>DO i = 1, n ..... ..... ..... END DO</pre>	<pre>DO i = 1, n, 2 ..... ..... ..... END DO</pre>	<pre>DO ..... IF (n &lt; 0) EXIT ..... END DO</pre>
---	--	---

## 演習6-1 DO構文の復習(2)

- 前回の演習5-1(点数を入力して、成績を出力するプログラム)を、**繰り返しの回数(n)を入力して、n回の成績処理**ができるよう変更しなさい。

```
PROGRAM point
INTEGER :: point
WRITE(*,*) '点数を入力してください'
READ(*,*) point
IF (point >= 80) THEN
WRITE(*,*) 'A(優)'
ELSE IF (point >= 70) THEN
WRITE(*,*) 'B(良)'
ELSE IF (point >= 60) THEN
WRITE(*,*) 'C(可)'
ELSE
WRITE(*,*) 'D(不可)'
END IF
STOP
END PROGRAM point
```

•nの入力は?



- 繰り返し処理は?
- どの処理を繰り返すか

## 演習6-2 DO構文の復習(3)

- 前回の演習5-1を、0以上の点数の入力に対して成績処理をし、負の数が入力されると処理を終了するよう変更しなさい。

```
PROGRAM point
INTEGER :: point
WRITE(*,*) '点数を入力してください'
READ(*,*) point
IF (point >= 80) THEN
WRITE(*,*) 'A(優)'
ELSE IF (point >= 70) THEN
WRITE(*,*) 'B(良)'
ELSE IF (point >= 60) THEN
WRITE(*,*) 'C(可)'
ELSE
WRITE(*,*) 'D(不可)'
END IF
STOP
END PROGRAM point
```



- 繰り返し処理は?
- どの処理を繰り返すか?
- 負の数が入力されたときに終了するには?

## DO構文の復習(4)

```
DO 記述しない
WRITE(*,*) '点数を入力してください'
READ(*,*) point
IF (point < 0) EXIT
IF (point >= 80) THEN
WRITE(*,*) 'A(優)'
.....
END IF
END DO
```



### 成績出力の処理(データがある間)

- 点数を入力して成績を出力する処理を繰り返すプログラムを作成する。入力データが終ると繰り返しを終了する。
  - 85 → A(優)
  - 63 → C(可)
  - 78 → B(良)
  - ☀ → 終了
- READ文では、入力待ち状態が続く。
- 入力データの終わりはどうやって表すか?
  - ◆ Ctrl-Dを入力

### 解法の説明

- 無限ループには、DO構文を使用
 

```
DO
...
END DO
```
- 入力データが終わると(Ctrl-Dを入力すると)DOループを抜ける(指定された行番号に制御が移る)
 

```
READ(*,*,END=10) a
...
10 CONTINUE
◆ CONTINUE文 何もしない実行文
```

### プログラムの一部

```
DO
WRITE(*,*) '点数を入力してください'
READ(*,*,END=10) point
IF (point >= 80) THEN
WRITE(*,*) 'A(優)'
.....
END IF
END DO
10 CONTINUE
```

END=10 入力データが終わったら、行番号10に制御を移す

### 演習6-3

- 点数を入力して成績を出力する処理を繰り返すプログラムを作成して実行しなさい。ただし、入力データが終ると(ctrl-Dが入力されると)繰り返しを終了する。

### 10進数から2進数への変換(1)

6 (10進数) → 110 (2進数)  
 13(10進数) → 1101 (2進数)

10進数 :  $n_k * 10^{k-1} + n_{k-1} * 10^{k-2} + \dots + n_2 * 10^1 + n_1 * 10^0$   
 2進数 :  $n_k * 2^{k-1} + n_{k-1} * 2^{k-2} + \dots + n_2 * 2^1 + n_1 * 2^0$

2 ) 6	...	0	2 ) 13	...	1
2 ) 3	...	1	2 ) 6	...	0
2 ) 1	...	1	2 ) 3	...	1
0	...	1	2 ) 1	...	1
			0	...	1

### 2進数への変換プログラム(1)

```
PROGRAM binary
INTEGER :: n
WRITE(*,*) '非負整数を入力してください'
READ(*,*) n
DO
WRITE(*,*) mod(n, 2)
n = n/2
IF (n = 0) exit
END DO
STOP
END PROGRAM binary
```

n = 6  
 mod(6,2) → 0 (出力)  
 n ← 6/2 (=3 ≠ 0)

n = 3  
 mod(3,2) → 1 (出力)  
 n ← 3/2 (=1 ≠ 0)

n = 1  
 mod(1,2) → 1 (出力)  
 n ← 1/2 (=0 = 0)

DO  
EXIT  
END DO

### 2進数への変換プログラムの問題点

- 逆向き
- 1行に表示されない  
非負整数を入力してください  
6  
0  
1  
1
- 出力する値を保存しておいて、逆順に出力  
1 1 0

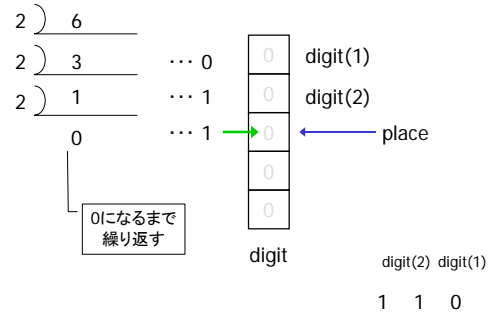
### 配列

- 一般の変数  
a      b
- 配列  
添字で指定  
a      a(1) a(2)      a(n)

### 配列の宣言

- 一般の変数の宣言  
INTEGER :: a,b
- 配列変数の宣言  
INTEGER :: a(1:20), b(1:20)  
INTEGER :: c(20), d(-5:5)  
INTEGER, DIMENSION(1:20) :: a, b  
※添字の下限は省略してよい(1となる)

### 10進数から2進数への変換(2)



### 2進数への変換プログラム(2)

```

.....
INTEGER :: n, i, place = 0, digit(1:32)
WRITE(*,*) '非負整数を入力してください'
READ(*,*) n
DO
    place = place + 1
    digit(place) = mod(n, 2)
    n = n/2
    IF (n = 0) exit
END DO
DO i = place, 1, -1
    WRITE(*,*) digit(i)
END DO
.....
    
```

### DO形並びによる出力

```

DO i = place, 1, -1
    WRITE(*,*) a(i)
END DO

WRITE(*,*)
digit(place), ..., digit(2), digit(1)

WRITE(*,*) (digit(i), i = place, 1, -1)
    
```

## DO形並び

- (値の並び, 変数 = 始値, 限界値, 増分値)

(digit(i), i = place, 1, -1)

(i\*\*2, i = 1, 10)

('#' , i = 1, 10)

```
WRITE(*,*) (i*j, j = 1, 9) ← DO i = 1, 9
                                DO j = 1, 9
                                WRITE(*,*) i*j
                                END DO
                                ENDO DO
```

## 演習6-4

- 結果の出力にDO形並びを使った2進数への変換プログラムを完成させなさい。
- 前回の9×9の一覧を、DO形並びを使って9×9の表を出力するように変更しなさい。

## 次回予定(第7回)

- ベクトル・行列の計算
- 書式制御
- 第一回レポートについて